UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,623	10/06/2003	Rene Rollig	5500-92201	3337

53806 7590 10/10/2007

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL (AMD)
P.O. BOX 398
AUSTIN, TX 78767-0398

EXAMINER
VIDWAN, JASJIT S

ART UNIT	PAPER NUMBER
2182	

MAIL DATE	DELIVERY MODE
10/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/679,623	ROLLIG ET AL.	
	Examiner	Art Unit	
	Jasjit S. Vidwan	2182	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 July 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 07/09/2007 have been fully considered but they are not persuasive. Applicant argues that prior art fails to teach:

(a) "Memory configured to store **microcode**"

(b) "Instruction fetch unit configured to **read** an instruction..."

(c) "Finite-state machine configured to receive **and interpret** the instructions..."

(d) "System Management Bus (SMBus) message handler

2. With respect to argument (a), **Examiner disagrees**. Applicant contests that plurality of programs stored in Luke's memory are written in high-level programming language (C) and not microcode as required by the claim limitation. Applicant also cited Wikipedia to highlight the difference between CPU instruction set and high-level programming language. However, it should be noted that though the program is written in a high-level programming language, once compiled the program stored on the memory is essentially instruction language set for the processor or controller to execute.

3. With respect to argument (b), **Examiner disagrees**. Applicant argues that the cited portion of Luke in Fig. 6, element 90 and further in Col. 7, Lines 17-19, does not read on the applicant's art as the instruction fetch unit fails to "read an instruction." However, Examiner believes that the Applicant had misinterpreted Examiner's citation of above element and passage for the limitation of "read an instruction." However, in the office action mailed 4/10/2007, Examiner points to Col. 9, Lines 14-20 which reads that the sequencer jumps to address specified in the memory device and executes the code that was present in the program. Therefore, it is the Examiner's position that "executing" an instruction reads on Applicant's claimed limitation of IFU configured to "**read** an instruction"

4. With respect to argument (c), **Examiner disagrees**. Applicant argues that the Finite-state machine is not configured to interpret instructions. However, in the cited portion of Luke's passage Col. 7, Lines 5-16, Luke teaches a State machine (82) which is coupled to the sequencer as shown above which reads and executes instructions of plurality of programs stored in the external memory (32). Luke further teaches the finite state machine (82) receiving signals and controlling the sequencer based on the said

signals determining properties such as number of times the sequencer should be executed. Therefore in light of the State machine being functionally coupled to the sequencer it would be inherent that the State machine would "read and interpret" the instructions from the programs as the state machine provides direction for the USB peripheral bridge.

5. With respect to argument (d), **Examiner disagrees**. In response to applicant's arguments, the recitation "SMBus message handler" has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) The invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1, 3-9, 23-19 and 21-31 are rejected under 35 U.S.C. 102(e) as being anticipated by Luke et al, U.S. Patent no: 6,505,267 **[herein after Luke]**.

3. **As per Claims 1,** Luke teaches a SMBus message handler **[see Fig. 2, element 16]** comprising:

(a) Memory **[See Fig. 2, element 32 & 40 – "external memory device" / "Sequencer RAM"]** configured to store microcode comprising at least two programs **[see Col. 7, Lines 58 – Col. 9, Line 12 – plurality of programs include 'Register read-modify-write', 'Register read-compare-until-match', 'Register Write', 'Register read extract nibble', 'Wait for bulk_in byte', Wait for bulk_out byte', 'DATI Push register into bulk_in', 'DATO Push bulk_out byte', 'EPPI Read EPP data register']** each for handling a bus command protocol and comprising at least one instruction **[see Col. 2, Lines 7-10 - - Also see Col. 4, Lines 66- Col. 5, Line 2]**.

(b) Interface **[Col. 4, Lines 17-21]** to a register **[see Fig. 3, element 66]** configured to identify a starting address of a program in said memory **[Col. 4, Lines 34-37]**

(c) Instruction fetch unit **[see Fig. 6, Element 90 – also see Col. 7, Lines 17-19]** configured to read an instruction at an address in said memory **[Col. 9, Lines 14-20]**, said address being specified by a program counter **[see Fig. 6, element 84]**

(d) Finite-state machine **[Fig. 6, element 82]** configured to receive and interpreting the instructions read by said instruction fetch unit **[Col. 7, Lines 5-16]** and for managing the data transfer between an SMBus interface, and a register set in compliance with said instructions read from said memory **[Col. 7, Lines 28-36]**.

4. **As per Claims 19**, Luke teaches method for controlling an SMBus:

(a) Identifying a starting address of a program **[Col. 4, Lines 34-37]** comprising one or more instructions, said program being stored in a memory **[See Fig. 2, element 32 & 40 – "external memory device" / "Sequencer RAM"]**

(b) Fetching instructions of said program **[Col. 7, Lines 17-19]** one after another **[see Fig. 6, element 84]** into a finite-state machine **[Fig. 6, element 82]**

(d) Transferring data between an SMBus interface and a register set **[Col. 7, Lines 5-16]** in compliance with the instruction present in said finite-state machine **[Col. 7, Lines 28-36]**.

5. **As per Claim 3 and 21,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising an address register array comprising a plurality of starting addresses of programs stored in said memory, said register comprising an offset for pointing at a specific register in said address register array **[Col. 4, Lines 31-42]**.

6. **As per Claims 6 and 28,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising a loop counter **[Fig. 6, element 88]** for storing the value of a block counter register in said loop counter if the finite-state machine executed a transmit data from block counter register instruction **[Col. 9, Lines 5-7, "DATO Push bulk_out byte into register"]**; said loop counter being decremented each time a data byte is transmitted to said SMBus interface while a "transmit data from" instruction is executed and the

"transmit data from" instruction be completed when the value of said loop counter reaches zero **[Col. 7, Lines 43-51]**.

7. **As per Claims 7 and 29,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising a loop counter **[88]** and a block counter register **[66]** both for storing a byte received from said SMBus interface if the finite-state machine **[82]** executed a "receive data to block counter register" instruction **[Col. 9, Lines 8-11],** said loop counter **[204]** being decremented each time a data byte is transmitted to or received from said SMBus interface while a "received data to block counter register " instruction is executed and the "received data to" instruction being completed when the value of said loop counter reaches zero.

8. **As per Claims 8 and 30,** Luke teaches SMBus message handler, wherein each instruction comprises one bit indicating as to whether or not an instruction is the last instruction in the program **[Col. 5, Lines 2-7]**.

9. **As per Claims 9 and 31,** Luke teaches SMBus message handler, wherein each instruction comprises one bit indicating as to whether an instruction is to be executed only once or this instruction is to be executed repeatedly until a loop counter becomes zero, wherein said loop counter is decremented each time an instruction is executed repeatedly **[Col. 7, Lines 38-57].**

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 2, 4, 5, 20, 22-27 rejected under 35 U.S.C. 103(a) as being unpatentable over Luke and further in view of Applicant Admitted Prior Art (Description of prior art) herein after [AAPA].

12. **As per Claims 2 and 20**, Luke teaches the above limitations of claims 1, 10 and 10. However, Luke fails to teach a system wherein the register set complies with the ACPI specification. AAPA teaches the above

deficiency of having a system wherein the register set is ACPI compliant **[see AAPA, Page 7, Paragraph 2 – Page 9, Paragraph 3]**.

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the teachings of Luke with that of AAPA in order to take advantage of a more efficient power management interface with regards to the register set. It is for this reason that one of ordinary skill in the art at the time of Applicant's invention would have been motivated to combine the teachings of Luke with that of AAPA in order to take advantage of a more efficient power management interface with regards to the register set.

13. **As per Claim 4 and 22**, Luke as modified by AAPA above teaches SMBus message handler [Fig. 2, element 16] further comprising a buffer pointer register **[Fig. 6, element 92]** for pointing at one of a plurality of data registers **[Fig. 3, element 66]**; said finite state machine **[Fig. 6, element 82]** transferring data read from SMBus interface to the data register at which said buffer pointer register points if said finite-state machine interprets a "receive data to" instruction; said finite state machine transferring the data read from the data register at which said buffer pointer register points to **[Col. 7, Line 58-65]** said SMBus interface if said finite-state machine interprets a "transmits data from" instruction **[Col. 8, Line 66-Col.9, Line 4]**

14. **As per Claims 5, 23 and 25** Luke as modified by AAPA above teaches SMBus message handler wherein the finite-state machine causes said buffer pointer register to be incremented each time a "transmit data to" or a "transmit data from" instruction is executed **[Col. 7, Lines 52-57]**

15. **As per Claims 24 and 27,** Luke as modified by AAPA above teaches a method wherein said transferring step further comprising decrementing a loop counter and checking as to whether said loop counter has a value of zero **[Col. 8, Lines 3-13]**.

16. **As per Claim 26,** Luke as modified by AAPA above teaches a method wherein said transferring step further comprising incrementing of said buffer pointer register **[Col. 7, Lines 44-50]**

17. Claims 10, 12, 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Luke et al and further in view of Newton's Telecom Dictionary **[herein Newton]**.

18. **As per Claim 10**, Luke teaches a system **[Bridge circuit]** for transmitting and receiving data over SMBus comprising:

(a) Interface to memory **[See Fig. 2, element 32 & 40 – "external memory device" / "Sequencer RAM"]** configured to store microcode comprising at least two programs **[see Col. 7, Lines 58 – Col. 9, Line 12 – plurality of programs include 'Register read-modify-write', 'Register read-compare-until-match', 'Register Write', 'Register read extract nibble', 'Wait for bulk_in byte', Wait for bulk_out byte', 'DATI Push register into bulk_in', 'DATO Push bulk_out byte', 'EPPI Read EPP data register']** each for handling a bus command protocol and comprising at least one instruction **[see Col. 2, Lines 7-10 - - Also see Col. 4, Lines 66- Col. 5, Line 2]**.

(b) Interface **[Col. 4, Lines 17-21]** to a register **[see Fig. 3, element 66]** configured to identify a starting address of a program in said memory **[Col. 4, Lines 34-37]**

(c) Instruction fetch unit **[see Fig. 6, Element 90 – also see Col. 7, Lines 17-19]** configured to read an instruction at an address in said memory **[Col. 9, Lines 14-20]**, said address being specified by a program counter **[see Fig. 6, element 84]**

(d) Finite-state machine **[Fig. 6, element 82]** configured to receive and interpreting the instructions read by said instruction fetch unit **[Col. 7, Lines 5-16]** and for managing the data transfer between an SMBus interface, and a register set in compliance with said instructions read from said memory **[Col. 7, Lines 28-36]**.

Luke teaches the above limitations, however fails to teach a system wherein the said bridge circuit is a integrated circuit chip. However, Newton teaches the benefit of having the above system on a chip **[see Newton, "System-on-chip"]**. One of ordinary skill in the art at the time of Applicant's invention would have been motivated to combine the two teachings in order to dramatically lower power, cost and real estate **[see Newton, Page 778, Paragraph 1 – 'SOC']**. It is for this reason that one of ordinary skill in the art at the time of Applicant's invention would have been motivated to combine the two teachings in order to dramatically lower power, cost and real estate **[see Newton, Page 778, Paragraph 1 – 'SOC']**.

19. **As per Claim 12,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising an address register array comprising a plurality of starting addresses of programs stored in said memory, said

register comprising an offset for pointing at a specific register in said address register array **[Col. 4, Lines 31-42]**.

20. **As per Claims 15,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising a loop counter **[Fig. 6, element 88]** for storing the value of a block counter register in said loop counter if the finite-state machine executed a transmit data from block counter register instruction **[Col. 9, Lines 5-7, "DATO Push bulk_out byte into register"]**; said loop counter being decremented each time a data byte is transmitted to said SMBus interface while a "transmit data from" instruction is executed and the "transmit data from" instruction be completed when the value of said loop counter reaches zero **[Col. 7, Lines 43-51]**.

21. **As per Claims 16,** Luke teaches SMBus message handler **[Fig. 2, element 16]** further comprising a loop counter **[88]** and a block counter register **[66]** both for storing a byte received from said SMBus interface if the finite-state machine **[82]** executed a "receive data to block counter register" instruction **[Col. 9, Lines 8-11],** said loop counter **[204]** being decremented each time a data byte is transmitted to or received from said SMBus interface while a "received data to block counter register " instruction is executed and the "received data to" instruction being completed when the value of said loop counter reaches zero.

22. **As per Claims 17,** Luke teaches SMBus message handler, wherein each instruction comprises one bit indicating as to whether or not an instruction is the last instruction in the program **[Col. 5, Lines 2-7]**.

23. **As per Claims 18,** Luke teaches SMBus message handler, wherein each instruction comprises one bit indicating as to whether an instruction is to be executed only once or this instruction is to be executed repeatedly until a loop counter becomes zero, wherein said loop counter is decremented each time an instruction is executed repeatedly **[Col. 7, Lines 38-57].**

24. Claims 11, 13 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Luke et al and Newton's Telecom Dictionary **[herein after Newton]** and further in view of Applicant Admitted Prior Art (background) **[herein after AAPA]**.

25. **As per Claims 11**, Luke and Newton teach the above limitations of claims 10. However, Luke fails to teach a system wherein the register set complies with the ACPI specification. AAPA teaches the above

deficiency of having a system wherein the register set is ACPI compliant **[see AAPA, Page 7, Paragraph 2 – Page 9, Paragraph 3]**.

It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the teachings of Luke with that of AAPA in order to take advantage of a more efficient power management interface with regards to the register set. It is for this reason that one of ordinary skill in the art at the time of Applicant's invention would have been motivated to combine the teachings of Luke with that of AAPA in order to take advantage of a more efficient power management interface with regards to the register set.

31. **As per Claim 13**, Luke and Newton as modified by AAPA above teaches SMBus message handler [Fig. 2, element 16] further comprising a buffer pointer register **[Fig. 6, element 92]** for pointing at one of a plurality of data registers **[Fig. 3, element 66]**; said finite state machine **[Fig. 6, element 82]** transferring data read from SMBus interface to the data register at which said buffer pointer register points if said finite-state machine interprets a "receive data to" instruction; said finite state machine transferring the data read from the data register at which said buffer pointer register points to **[Col. 7, Line 58-65]** said SMBus interface if said finite-state machine interprets a "transmits data from" instruction **[Col. 8, Line 66-Col.9, Line 4]**

32. As per Claims 14 Luke and Newton as modified by AAPA above teaches SMBus message handler wherein the finite-state machine causes said buffer pointer register to be incremented each time a "transmit data to" or a "transmit data from" instruction is executed **[Col. 7, Lines 52-57]**

Conclusion

26. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action: In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jasjit S. Vidwan whose telephone number is (571) 272-7936. The examiner can normally be reached on 8am - 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, KIM HUYNH can be reached on (571) 272-4147. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JSV
9/28/07

9/29/07